

REMARKS

The abstract has been amended to make editorial changes therein, bearing in mind the criticisms in the Official Action, to place the application in condition for allowance at the time of the next Official Action.

The form of the claims has been amended as requested in the Official Action. In addition, for clarity reasons, in claim 1 the expression "a remote database operation" is amended to be in the form "a remote transaction" to point out that a remote database operation is a third transaction in the transaction scheme, as shown in Figures 3b and 6 (with reference 33) and 5 (with references 522, 532, 542) according to the invention. This amendment is also justified by the definition given in the description on page 2, lines 20 and 29 to 30.

Claim 26 is amended for clarity reasons by replacing the feature "push data into" with the above feature "invoke a remote transaction". The basis for this amendment is the same as in claim 1, and also in the description on page 10, lines 1 to 8.

For clarity reasons claims 13 and 34 are amended by adding a feature "in the first database" to point out that a second transaction is initiated in the first database to synchronize data in the second database. The basis for this clarification can be found in the description on page 10, lines 16 to 17.

Claims 1-9, 11-21, 23-29, 32-38 and 41-42 were rejected as anticipated by ORAREP (Oracle7™ Server Distributed Systems, Volume II: Replicated Data, Release 7.3, Volume II, February 1996, Part No. A32545-2, ORACLE®). The claims have been amended and reconsideration and withdrawal of the rejection are respectfully requested.

ORAREP discusses mostly asynchronous replication in distributed database systems and to some extent synchronous and procedural replicated used in specialized applications. ORAREP teaches that only procedural replication can be run serially in conjunction with synchronous row-level replication (page 1-16).

According to ORAREP a database object (replicated object) is copied to multiple sites in the distribution system. Snapshot sites only push/pull down changes to/from their associated master sites. A master definition site is required as the control point for performing schema-level changes and administrative activities (by calling packaged procedures). These aspects are discussed in ORAREP introduction from page 1-2 to 1-7.

ORAREP teaches that multi-master replication supports full-table replication between master tables. Changes applied to any master table are propagated directly either synchronously or asynchronously to all other master tables (page 1-8) in the replicated environment. In hybrid configuration also replication of subsets of master table data is supported (page 1-10). When

updating an object the change is propagated to all master sites and snapshot sites.

Thus, it would be axiomatic that in ORAREP a data operation completed in a second site would be the same data operation as completed in a first site. It is unthinkable to change a data operation during this procedure.

For multiple master replication ORAREP generates a trigger and stored procedure for that table to support the replication of data-level changes. When performing a change locally, ORAREP fires a generated trigger that builds a remote procedure call to a packaged procedure at the remote site. These procedures contain information to apply the change at the remote site. These procedures contain information to apply the change at the remote site. This asynchronous (deferred transactions) mechanism is shown in Figure 1-6 and page 1-12 in ORAREP. Figure 1-7 shows a synchronous mechanism in which making a change to a replicated table ORAREP fires a trigger which calls a package procedure at each master site that applies the change. ORAREP generates the packages in two phases.

ORAREP Figure 4-1 shows that in the synchronous row-level replication a change to a local table is propagated to other master sites using generated triggers and their associated packages. Columns cannot be replicated. When the local change is applied these triggers issue calls to generated procedures at the remote master sites (page 4-23). The necessary remote

procedure calls to support synchronous replication are included in the generated trigger and its associated package for each object.

The generated trigger according to ORAREP is a database operation level trigger which is connected to a single database operation, such as insert, remove and update. This kind of trigger has been known for more than 20 years. It is possible to define such a trigger that always fires whenever a row is added to a table. This trigger can check whether the added row is valid or it can update data to another table according to the data of the added row. The trigger is committed in the same transaction as was committed the database operation that fired said trigger. The generated trigger described in Figure 1-6 and on page 1-12 as well as on page 4-23 is the trigger of this kind. This system generated trigger stores committed operations to the transaction queue table from where they are later replicated to other site automatically by means of some remote procedure. I.e., always when a row is added to a table, the automatically generated trigger takes care that a transaction queue table is updated with a corresponding row using replication (EMP row and package procedure calls in ORAREP Figures 1-6 and 1-7).

The provisions for serial execution are restricted as now discussed below. Procedural replication mechanism, which can be stored in specialized cases, only replicates the call to a stored procedure that is used to update a table. Procedural

replication does not replicate the update itself (page 1-15). As early mentioned, ORAREP teaches that pocedural replication can be run serially in conjunction with synchronous row-level replication (page 1-16). It suits for e.g. purging deleted rows in database. There are listed a lot of restrictions on procedural replication on pages 8-13 and 8-14, e.g. regarding to e.g. deferred procedures. Procedural replication uses procedure wrappers (package procedure in a new wrap) to build deferred transactions (pages 4-26, 8-17), but then you have to disable row-level replication support (page 8-15).

ORAREP describes on page 8-14 a serial execution of transactions A and B performing updates on local data. First executing transactions A and B serially locally (in the first database) and after committing transactions A and B locally (in the first database), the replicas of A an B on other sitse (in the second database) are executed serially, in the same order that they were committed in the first database. If A and B are executed concurrently locally there is needed complicated procedures to ensure consistency.

ORAREP does not disclose a transaction trigger including attributes is linked into said first transaction.

In ORAREP there is not defined any transaction trigger in the first database to be linked to the first transaction. Indeed, this step is absent from ORAREP, because a data operation

committed in the first and second databases is absolutely the same data operation.

The claimed method provides that the application programmer can freely program attributes in the transaction trigger to determine what will happen in step (v) later on. In other words, the programmed transaction trigger inside the first transaction will tell the system that after this first transaction is completed in the first database, the second transaction is initiated in the first database and the third transaction (procedure) shall be completed in the second database according to the information on the transaction trigger. Grounds for this can be found from description on page 9, lines 12 to 14. The trigger of the invention is a transactional level trigger defined by the programmer and it fires when the whole first transaction, not only a single operation of it, is successfully committed.

A trigger according to ORAREP is a database operation level trigger which is connected to a single database operation, such as insert, remove and update. This trigger is committed in the same transaction as was committed the database operation that fired said trigger. The generated trigger describes in Figure 1-6 and on page 1-12 as well as on page 4-23 is the trigger of this kind. This system generated trigger stores committed operations to the transaction queue table from where they are later replicated to other site automatically by means of some

remote procedure as described earlier in the section of the prior art.

ORAREP also does not disclose that a remote transaction is invoked in a second database according to at least some of the attributes in the trigger.

ORAREP does not teach that a third transaction is invoked in the second database after the first transaction is completed. ORAREP teaches that the same transaction that was run in the site A is automatically replicated in the site B as well (Figure 4-1 and text on page 4-23). In the present invention, the programmed transaction trigger including attributes inside the first transaction will tell the system that after this first transaction is completed in the first database, the third transaction shall be completed in the second database according to the information in the transaction trigger (some attributes). In other words the second database completes a procedure whose call with parameters is defined in the transaction trigger that was fired in the first database. For example this procedure completed in the second database may request the first database to transmit changed data to itself.

Accordingly, these claims avoid the rejection under §102.

Claims 10, 22, 30-31 and 39-40 were rejected as unpatentable over ORAREP in view of ORANET (Oracle® Advanced Networking Option™, Administrator's Guide, Release 2.3.3, Part

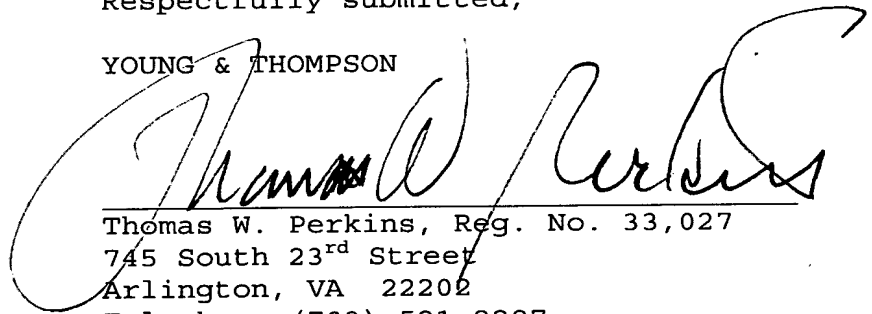
No. A48511-1, ORACLE®, 1996). Reconsideration and withdrawal of the rejection are respectfully requested for the reasons given above.

In view of the present amendment and the foregoing remarks, it is believed that the present application has been placed in condition for allowance. Reconsideration and allowance are respectfully requested.

The Commissioner is hereby authorized in this, concurrent, and future replies, to charge payment or credit any overpayment to Deposit Account No. 25-0120 for any additional fees required under 37 C.F.R. § 1.16 or under 37 C.F.R. § 1.17.

Respectfully submitted,

YOUNG & THOMPSON

A large, stylized handwritten signature in black ink, appearing to read 'Thomas W. Perkins', is written over the printed name and address.

Thomas W. Perkins, Reg. No. 33,027  
745 South 23<sup>rd</sup> Street  
Arlington, VA 22202  
Telephone (703) 521-2297  
Telefax (703) 685-0573  
(703) 979-4709

TWP/fb



**APPENDIX:**

The Appendix includes the following item(s):

- ☐ - a terminal disclaimer
- ☐ - a 37 CFR 1.132 Declaration
- ☒ - a new or amended Abstract of the Disclosure
- ☐ - a Replacement Sheet for Figure                      of the drawings
- ☐ - a Substitute Specification and a marked-up copy of the  
originally-filed specification
- ☐ - a verified English translation of foreign priority document